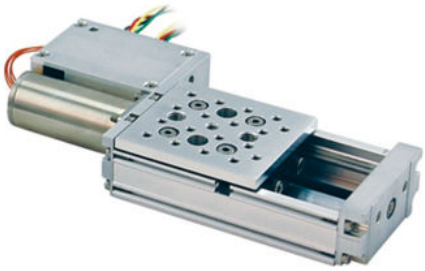
# ZABER

T-LSM-SV Product
User's Manual
Firmware 5.00 and up

Last Update: December 10 2015
Visit www.zaber.com/wiki for more recent updates.

Zaber Technologies Inc.
#2 - 605 West Kent Ave. N.
Vancouver, British Columbia
Canada, V6P 6T7



# Table of Contents

**Disclaimer** ..... 1

**Precautions** ..... 2
- Maintenance ..... 2

**Vacuum Device Handling Guidelines** ..... 3

**Initial Setup and Testing** ..... 4

**Installation** ..... 5
- Connecting Vacuum Devices to a Computer ..... 5
  - Power ..... 5
  - Wiring/Feed-throughs ..... 5
  - Vacuum Data Cable Wiring Diagram ..... 5
  - Grounding ..... 6
  - Connect the device to a computer as follows: ..... 6
- Physical Installation ..... 7
  - Mounting ..... 7
  - Grounding ..... 7
  - Vacuum Device Mounting Considerations ..... 7

**Temperature Monitoring** ..... 8

**Vacuum Baking** ..... 9

**Current Settings** ..... 10
- Running Current ..... 10
- Hold Current ..... 10

**Control Through The RS-232 Serial Port** ..... 11
- Examples ..... 11
- Data Conversion Algorithms ..... 12
- Sample Waveforms ..... 12

**Quick Command Reference** ..... 14

**Detailed Command Reference** ..... 16
- Reset - Cmd 0 ..... 16
  - Special Note ..... 16
- Home - Cmd 1 ..... 16
- Renumber - Cmd 2 ..... 17
- Store Current Position - Cmd 16 ..... 18
- Return Stored Position - Cmd 17 ..... 18
- Move To Stored Position - Cmd 18 ..... 18
- Move Absolute - Cmd 20 ..... 19
- Move Relative - Cmd 21 ..... 20
- Move At Constant Speed - Cmd 22 ..... 20
- Stop - Cmd 23 ..... 21

# Table of Contents

**Detailed Command Reference**
- Read Or Write Memory - Cmd 35 ..... 22
- Restore Settings - Cmd 36 ..... 22
- Set Microstep Resolution - Cmd 37 ..... 23
- Set Running Current - Cmd 38 ..... 24
- Set Hold Current - Cmd 39 ..... 25
- Set Device Mode - Cmd 40 ..... 26
- Set Home Speed - Cmd 41 ..... 29
- Set Target Speed - Cmd 42 ..... 30
- Set Acceleration - Cmd 43 ..... 31
- Set Maximum Position - Cmd 44 ..... 32
- Set Current Position - Cmd 45 ..... 33
- Set Maximum Relative Move - Cmd 46 ..... 33
- Set Home Offset - Cmd 47 ..... 34
- Set Alias Number - Cmd 48 ..... 35
- Set Lock State - Cmd 49 ..... 35
- Return Device Id - Cmd 50 ..... 36
- Return Firmware Version - Cmd 51 ..... 36
- Return Power Supply Voltage - Cmd 52 ..... 37
- Return Setting - Cmd 53 ..... 37
- Return Status - Cmd 54 ..... 38
- Echo Data - Cmd 55 ..... 38
- Return Current Position - Cmd 60 ..... 39

**Reply-Only Reference ..... 40**
- Move Tracking - Cmd 8 ..... 40
- Limit Active - Cmd 9 ..... 40
- Manual Move Tracking - Cmd 10 ..... 41
- Error - Cmd 255 ..... 41
  - Error Codes ..... 42

**Manual Control ..... 44**

**LED functions ..... 45**

**Troubleshooting T-Series Motion Devices ..... 46**

**Warranty and Repair ..... 50**
- Standard products ..... 50
- Custom products ..... 50
- How to return products ..... 50

**Email Updates ..... 51**

**Contact Information ..... 52**

# Table of Contents

**Appendix A: Default Settings**.....53

**Appendix B: Device Specifications**.....54

# Disclaimer

Zaber’s devices are not intended for use in any critical medical, aviation, or military applications or situations where a product's use or failure could cause personal injury, death, or damage to property. Zaber disclaims any and all liability for injury or other damages resulting from the use of our products.

# Precautions

Zaber's linear stages can produce enough force to cause personal injury. Be careful to not get body parts caught between the moving carriage and the end plates of the stage. Make sure that no loose clothing gets caught in the rotating lead screw.

## Maintenance

Teflon-coated (black) lead screws are not designed to be greased. Normally no lubrication is required over the lifetime of the device. If a squeaking sound develops in the lead screw, Super Lube oil with PTFE (Product number #51004) can be applied. Apply a very thin bead over 80% of the contact surface of the lead screw and run the stage slowly from end to end. Wipe off any excess.

# Vacuum Device Handling Guidelines

Do not touch SV2 vacuum devices with your bare hands. The oil from your skin can significantly increase the pump down time of your chamber. Always wear cleanroom gloves when handling high vacuum positioners and other high vacuum components.

The T-LSM-SV stages have large vent holes in their motors and contains powerful magnets. Particles can easily be drawn into the motor and will greatly decrease stage performance. Keep metal particles away from the motor. Dirt, dust, humidity or other foreign particles that penetrate the motor housing can also degrade performance. Make sure the device is stored and operated only in clean environments.

# Initial Setup and Testing

If you are unfamiliar with T-series devices, you may wish to perform a few simple steps to familiarize yourself with their operation. First you will need a power supply. If you did not add one to your order, you will require a power supply that matches the voltage requirement of the device. Some T-series products, usually devices that run on 12 - 16 V, transmit power between devices through the data cables. The chosen power supply must be rated to handle the maximum total current draw of all devices connected to it. When powering long chains of devices, we recommend connecting a power supply to at least every 3rd device in the chain to reduce the current through the data cables. Devices that run on more than 16 V do not transmit power through the data cables.

Once you have a working power supply, you can test the operation of your device. Try turning the potentiometer (if applicable) counterclockwise to home the device. When operating manually, you should home the device by retracting it completely every time you disconnect or turn off the power. You will not be able to achieve the maximum travel until you have homed the device or sent a command to tell the device its current position. After the device retracts completely it will stop automatically and you can turn the potentiometer clockwise to move the device. If you have a model without a potentiometer, you cannot control the device manually. Instead you must connect the device to a computer and issue a home instruction.

# Installation

## Connecting Vacuum Devices to a Computer

Vacuum compatible devices have special requirements for their connection to a computer. Please read the instructions below carefully.

### Power

No power connectors are provided on vacuum compatible devices. Instead, power must be supplied through the data cables. A special serial adapter with power input, the T-DSUB9-P, is provided.

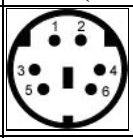
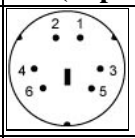
T-DSUB9-P Mini-DIN to D-Sub serial adapter

### Wiring/Feed-throughs

SV1 and SV2 vacuum devices are supplied with twisted signal wires and standard serial port connectors. The wires must be cut and soldered to a vacuum feed-through before the device can be operated inside a vacuum chamber.

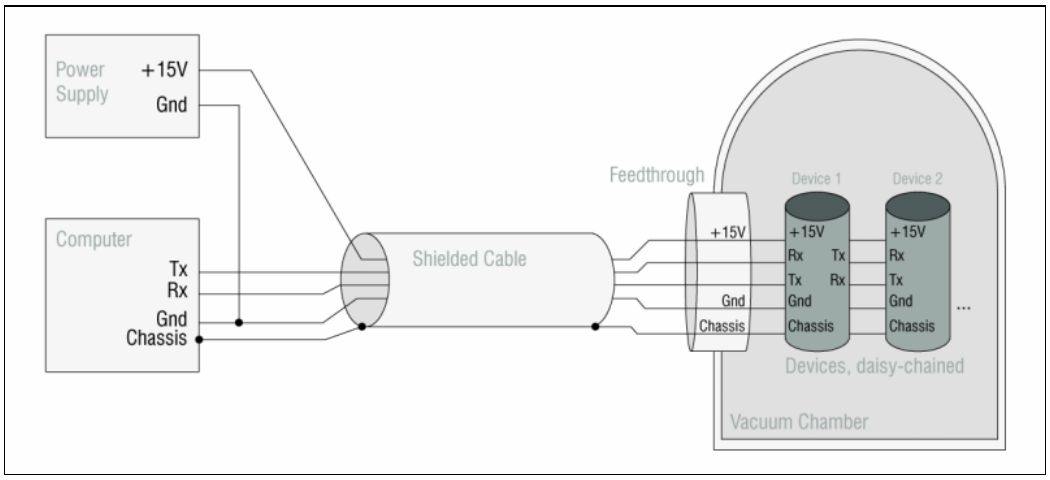
Two or three devices can be daisy chained inside a vacuum chamber, eliminating the need for extra feed-throughs. If this is to be done, it is recommended that the connectors be removed and the signal wires connected directly from one device to the next in the chain. The wires colours should be matched.

### Vacuum Data Cable Wiring Diagram

| Pin# | Minidin 6 Female (to next device) | Minidin 6 Male (to previous device) | Wiring Colour (applies to vacuum devices only) |
|---|---|---|---|
| |  |  | |
| 1 | - | - | - |
| 2 | Receive (from next device) | Transmit (toward computer) | Brown |
| 3 | Ground | Ground | Red |
| 4 | +12-16 Vdc * | +12-16 Vdc * | Orange |
| 5 | - | - | - |
| 6 | Transmit (to next device) | Receive (from computer) | Green |
| nc | - | - | - |

## Grounding

When re-wiring Zaber devices it is important to wire the ground to both the computer and the power supply (see diagram below). Alternately, the T-DSUB9-P vacuum adapter can be substituted into an order to make this connection.




Note that one can use an optional shielded cable (not included) outside the vacuum chamber to improve EMI performance. It not using a shielded cable, the CHASSIS line should still be connected to protect against ESD.

## Connect the device to a computer as follows:

1. Connect the Mini-DIN to D-Sub9 powered serial adapter (T-DSUB9-P) to your computer's serial port and plug the device's data cable into the adapter. You may need to use a cable extension to reach your computer. There is no need to power-down or reboot the computer. USB-to-RS232 converters are available if you have no RS232 serial port.
2. Connect the power plug of your power supply to the power connector on the T-DSUB9-P adapter. If present, the green LED should light up indicating the device has power (some devices may not have LEDs).
3. Additional devices can be daisy-chained to the first inside the vacuum chamber. A maximum of 3 vacuum devices can be connected to the same daisy chain. A renumber command needs to be issued after a new device is connected, before it can be controlled independently.
4. Install software from http://www.zaber.com/wiki/Software. We recommend installing the Zaber Console to get started. As a simple first test, try entering these instructions:

```
Renumber all devices               - Device: 0, Cmd:  2, Data: 0
Home device 1                      - Device: 1, Cmd:  1, Data: 0
Move device 1 to 10,000 microsteps  - Device: 1, Cmd: 20, Data: 10000
```

To see the microstep distance (resolution) for your device, see the specs page for that device on the respective product page .

# Physical Installation

## Mounting

There are several options available for mounting Zaber stages. Use the mounting holes in the bottom to mount to a surface or to another stage. You might have to move the carriage to access the bottom mounting holes. Some stages have mounting holes in the end plates for mounting vertically. Mounting screws are included with most stages.

**Caution:** Some stages have threaded through-holes in the top mounting plate of the carriage. Be sure not to install mounting screws too deep, causing them to interfere with inside parts of the stage.

T-LSM stages can be mounted to a standard metric or imperial breadboard with our AP101 adaptor plates.

## Grounding

To prevent damage to the device due to static buildup, the device should be properly grounded.

Failure to ground the unit may result in the unit shutting down unexpectedly or ceasing to communicate with the computer. This problem can be minimized by not touching the unit during operation. If the unit fails due to static discharge, unplugging it and plugging it back in or sending a Restore Settings command will usually fix the problem.

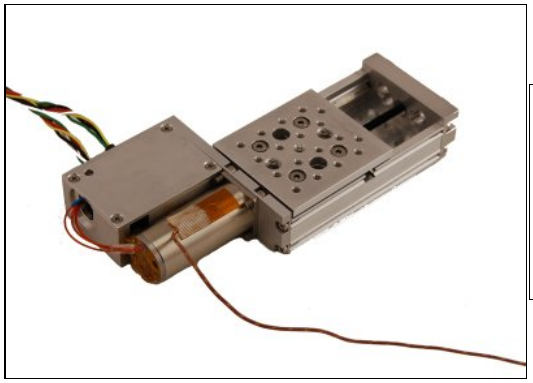
Most Zaber devices are grounded via the shield wire of the data cables. This should normally provide a path to ground via the computer. For units which are being used without a computer, a ground lead should be connected to the shield of one of the data cables.

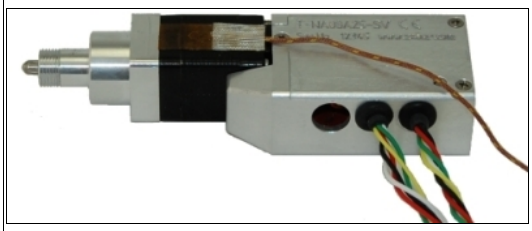
## Vacuum Device Mounting Considerations

If possible, vacuum positioners should be mounted to a thermally conductive surface to aid in head dissipation. Preferably, there should be a conductive heat path from the positioner through to the outside of the chamber. If this is not possible, the device should be mounted to a large conductive surface within the chamber. This will help cooling by increasing the rate of radiative heat transfer.

# Temperature Monitoring

A K-type thermocouple or other temperature sensor may be mounted to the stage to monitor temperature in order to reduce the risk of overheating. The thermocouple should be attached to the device's motor housing. A couple of examples of proper mounting locations are shown below.





The lifetime of the motor and stage will be prolonged by keeping the operating temperatures as low as possible. The motor housing temperature of the T-NA-SV and T-LSM-SV vacuum devices should never exceed 80°C.

# Vacuum Baking

Vacuum devices can be vacuum baked to reduce pump down time and outgassing. The procedure is to heat the device to its recommended bake-out temperature at a vacuum of less than 10^-3 Torr for a period of several hours. Allowable bake-out temperatures differ between devices. See the list below for the maximum bake-out temperature for your device:

**T-LSM-SV stages (SV1 and SV2):** Maximum bake-out temperature is 80°C

**T-NA-SV actuators (SV1 and SV2):** Maximum bake-out temperature is 110°C

Do not exceed these temperatures as there are polymer based components in the stage that can be permanently damaged by overheating.

# Current Settings

## Running Current

The default running currents on Zaber vacuum devices are set low enough that even prolonged operation will not damage the motor or controller. Due to the reduced heat transfer in vacuum chambers, this means that the thrust and speed performance of these devices is usually less than their non-vacuum counterparts.

It is possible to increase the running current of vacuum devices. This will increase the maximum speed and thrust of the device, but will also lead to increased heating. The maximum running current of the device is restricted to a level that will not damage the device under normal operating conditions. If the running current is to be increased above the default setting, it is recommended that the device be mounted to a thermally conductive surface that can transfer heat outside the vacuum chamber. It is also a good idea to minimize the duty cycle of the device. These steps will help extend the lifetime of the motor.

## Hold Current

It is good practice to reduce the hold current as much as is practical for vacuum devices. Positioners with an A pitch leadscrew often do not need a hold current at all to maintain their position. Reducing the hold current is an easy way to reduce the heat generated by your vacuum stage or actuator.

# Control Through The RS-232 Serial Port

All T-Series devices use the same RS232 communications protocol. Your communications settings must be: 9600 baud, no hand shaking, 8 data bits, no parity, one stop bit. The yellow LED will light when there is activity on the RS232 lines. You may use this feature to determine which COM port you are connected to. We recommend using the Zaber Console that you can download from our web site. The source code is also available for you to use as an example for writing your own custom code. See the troubleshooting section later in this manual if you have trouble communicating with the device.

**Important:** The first time you connect a device to your computer you must issue a renumber instruction to assign each device a unique identifier. This should be done after all the devices in the daisy-chain are powered up. In older firmware versions (prior to version 5xx) you must issue a renumber instruction after each powerup. In firmware 5xx and up, the device number is stored in non-volatile memory and will persist after powerdown, so you need only issue the renumber instruction when you add new devices to the chain, or rearrange the order of the devices, however it does no harm to issue the renumber instruction after every powerup. You must not transmit any instructions while the chain is renumbering or the renumbering routine may be corrupted. Renumbering takes less than a second, after which you may start issuing instructions over the RS232 connection.

All instructions consist of a group of 6 bytes. They must be transmitted with less than 10 ms between each byte. If the device has received less than 6 bytes and then a period longer than 10 ms passes, it ignores the bytes already received. We recommend that your software behave similarly when receiving data from the devices, especially in a noisy environment like a pulsed laser lab.

The following table shows the instruction format:

- Byte 1 - Device #
- Byte 2 - Command #
- Byte 3 - Data - Least Significant Byte (LSB)
- Byte 4 - Data
- Byte 5 - Data
- Byte 6 - Data - Most Significant Byte (MSB)

The first byte is the device number in the daisy-chain. Device number 1 is the closest device to the computer and device number 2 is next and so on. If the number 0 is used, all the devices in the chain will process the accompanying command simultaneously.

The second byte is the command number. Bytes 3, 4, 5, and 6 are data in long integer, 2's complement format with the least significant byte transmitted first. How the command data are interpreted depends on the command. Complete details are given in the command reference on the following page.

## Examples

- All devices renumber: 0, 2, 0, 0, 0, 0
- All devices home: 0, 1, 0, 0, 0, 0
- All devices return firmware version: 0, 51, 0, 0, 0, 0
- Device 1 move to an absolute position (command 20) of 257 microsteps: 1, 20, 1, 1, 0, 0
- Device 2 move to a relative position (command 21) of -1 microstep: 2, 21, 255, 255, 255, 255

Most instructions cause the device to reply with a return code. It is also a group of 6 bytes. The first byte is the device #. Byte #2 is the instruction just completed or 255 (0xFF) if an error occurs. Bytes 3, 4, 5 and 6 are data bytes in the same format as the instruction command data.

# Data Conversion Algorithms

If you are writing software to control Zaber products, you'll likely need to generate data bytes 3 through 6 from a single data value, or vise versa. The following pseudo-code can be used as a model.

**Converting command data into command bytes to send to Zaber products**

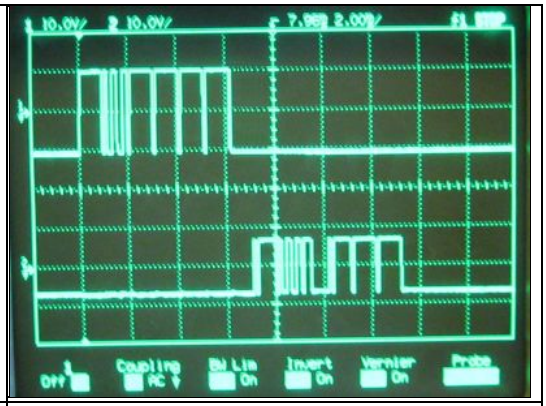
```
If Cmd_Data < 0 then Cmd_Data = 256^4 + Cmd_Data                    '
Cmd_Byte_6 = Cmd_Data / 256^3
Cmd_Data   = Cmd_Data - 256^3 * Cmd_Byte_6
Cmd_Byte_5 = Cmd_Data / 256^2
Cmd_Data   = Cmd_Data - 256^2 * Cmd_Byte_5
Cmd_Byte_4 = Cmd_Data / 256
Cmd_Data   = Cmd_Data - 256   * Cmd_Byte_4
Cmd_Byte 3 = Cmd_Data
```

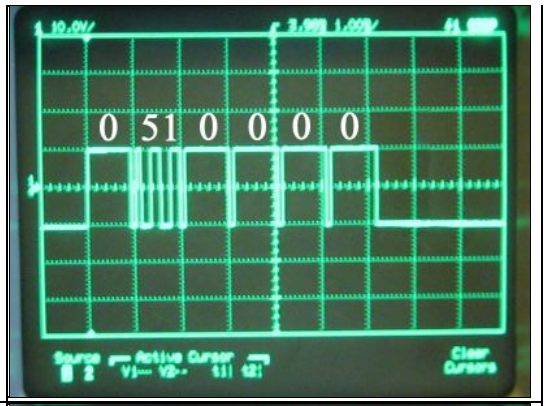
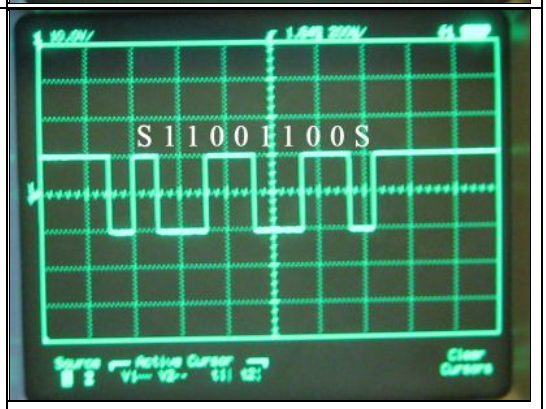
**Converting reply bytes into a single reply data value**

```
Reply_Data = 256^3 * Rpl_Byte 6 + 256^2 * Rpl_Byte_5 + 256 * Rpl_Byte_4 + Rpl_Byte_3
If Rpl_Byte_6 > 127 then Reply_Data = Reply_Data - 256^4            '
```

# Sample Waveforms

If you are designing hardware to interface with Zaber products, it may be useful to see some sample waveforms.

| | |
|---|---|
| This image shows an instruction (top waveform) and a reply (bottom waveform). |  |
| This is a closeup of the instruction bytes. Note that the voltage swing of the instruction waveform is about +/-10V. This is typical of the USB to RS232 converter to which the device is connected. The instruction bytes are 0, 51, 0, 0, 0, 0 indicating the instruction: Device: All, Command: Return version, Data: 0. | |

| | |
|---|---|
| |  |
| This is a closeup of the instruction byte 2. The "S" on either end represents the start and stop bits respectively. The start bit is always positive and the stop bit is always negative. The bit sequence is Start, 0, 1, 2, 3, 4, 5, 6, 7, Stop. The data is output with the least significant bit first, so the byte shown is actually 00110011 in binary or 32+16+2+1 = 51 in decimal.<br><br>It's a good idea to do a timing "reality check" whenever viewing a waveform such as this. The bit rate should be 9600 bits/second. The time scale in the image is 200 us/division. Since a byte consists of 10 bits (start, stop, plus 8 data bits) and the bit rate is 9600 bits/s, the duration of a single byte should be 1.04 ms (10/9600 seconds) or about 5.2 divisions at 200 us/division. It can be seen in the image that this is the case. |  |
| This is a closeup of the reply bytes. Note that the voltage swing of the reply waveform is about +/-8V. This is a typical output from the RS232 driver chip used in Zaber devices. The instruction was to all devices. Only a single device is present so there is only one reply. The reply is 1, 51, 252, 1, 0, 0 indicating the reply "Device: 1, Reply: Firmware version, Data: 508". The data is the last 4 bytes combined, with least significant byte first. In this case the data is 0*256^3 + 0*256^2 + 1*256 + 252 = 508. This indicates a firmware version of 5.08 since according to the instruction specification a decimal is assumed before the last two digits.<br><br>The time scale in this image is 1 ms/division. As noted above, at a bit rate of 9600 bits/second each byte should take 1.04 ms. Therefore a 6 byte reply should take 6.24 ms. It can be seen in the image that this is the case. | |

# Quick Command Reference

The following table offers a quick command reference for motorized devices running firmware version 5xx. For convenience, you may sort the table below by instruction name, command number, or reply number. Follow the links to view a detailed description of each instruction.

| Instruction Name | Command# | Command Data | Command Type | Reply Data |
|---|---|---|---|---|
| Reset | 0 | Ignored | Command | None |
| Home | 1 | Ignored | Command | Final position (in this case 0) |
| Renumber* | 2 | Ignored | Command | Device Id |
| Move Tracking | 8 | n/a | Reply | Tracking Position |
| Limit Active | 9 | n/a | Reply | Final Position |
| Manual Move Tracking | 10 | n/a | Reply | Tracking Position |
| Store Current Position* | 16 | Address | Command | Address |
| Return Stored Position | 17 | Address | Command | Stored Position |
| Move To Stored Position | 18 | Address | Command | Final Position |
| Move Absolute | 20 | Absolute Position | Command | Final Position |
| Move Relative | 21 | Relative Position | Command | Final Position |
| Move At Constant Speed | 22 | Speed | Command | Speed |
| Stop | 23 | Ignored | Command | Final Position |
| Read Or Write Memory* | 35 | Data | Command | Data |
| Restore Settings* | 36 | Peripheral Id | Command | Peripheral Id |
| Set Microstep Resolution* | 37 | Microsteps | Setting | Microsteps |
| Set Running Current* | 38 | Value | Setting | Value |
| Set Hold Current* | 39 | Value | Setting | Value |
| Set Device Mode* | 40 | Mode | Setting | Mode |
| Set Home Speed* | 41 | Speed | Setting | Speed |
| Set Target Speed* | 42 | Speed | Setting | Speed |
| Set Acceleration* | 43 | Acceleration | Setting | Acceleration |
| Set Maximum Position* | 44 | Range | Setting | Range |
| Set Current Position | 45 | New Position | Setting | New Position |
| Set Maximum Relative Move* | 46 | Range | Setting | Range |
| Set Home Offset* | 47 | Offset | Setting | Offset |
| Set Alias Number* | 48 | Alias Number | Setting | Alias Number |
| Set Lock State* | 49 | Lock Status | Command | Lock Status |
| Return Device Id | 50 | Ignored | Read-Only Setting | Device Id |
| Return Firmware Version | 51 | Ignored | Read-Only Setting | Version |

| | | | | |
|---|---|---|---|---|
| Return Power Supply Voltage | 52 | Ignored | Read-Only Setting | Voltage |
| Return Setting | 53 | Setting Number | Command | Setting Value |
| Return Status | 54 | Ignored | Read-Only Setting | Status |
| Echo Data | 55 | Data | Command | Data |
| Return Current Position | 60 | Ignored | Read-Only Setting | Position |
| Error | 255 | n/a | Reply | Error Code |

* The settings for these commands are saved in non-volatile memory, i.e. the setting persists even if the device is powered down. To restore all settings to factory default, use command 36.

# Detailed Command Reference

This command reference applies only to firmware version 5.00 and up to the most recent version. The version of firmware installed on any Zaber T-Series device can be determined by issuing command #51. A three-digit number will be returned. Assume 2 decimal places (ex a reply of 293 indicates firmware version 2.93). For earlier versions of firmware, please consult the appropriate PDF user's manual:

T-Series Manual 2xx (PDF)
T-Series Manual 4xx (PDF)

Due to the addition of new features, newer versions of firmware may not be 100% backward compatible. You may wish to read the document Firmware History and Migration which outlines the changes that have taken place from one firmware version to the next and indicates what options are available if you wish to upgrade or downgrade the firmware on your devices.

## Reset - Cmd 0

| | |
|---|---|
| **Instruction Name** | Reset |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 0 |
| **Command Type** | Command |
| **Command Data** | Ignored |
| **Reply Data** | None |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Sets the device to its power-up condition. |

This has the same effect as unplugging and restarting the device.

### Special Note

The position stored in the device prior to this command will be lost, so you must save it and reload it if it is important. All non-volatile settings (i.e. Device Number, Target Velocity, etc.) are saved and are not affected by reset or power-down.

## Home - Cmd 1

| | |
|---|---|
| **Instruction Name** | Home |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 1 |

| | |
|---|---|
| **Command Type** | Command |
| **Command Data** | Ignored |
| **Reply Data** | Final Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Moves to the home position and resets the device's internal position. |

Upon receiving this instruction, the device will retract until its internal home sensor is triggered. It will then move forward several steps to avoid accidentally re-triggering the home sensor during use. Its internal position is then reset (to 0 for most devices). If a home offset has been specified with the Set Home Offset (cmd 47) instruction, the device will move forward for the specific offset, then reset the internal position.

Prior to Firmware 5.21, the device will attempt to home for an extended amount of time. For Firmware 5.21 and up, the home command aborts with an error if the device has traveled twice the Maximum Position setting without triggering the home sensor. This indicates that the device could possibly be stalling or slipping.

# Renumber - Cmd 2

| | |
|---|---|
| **Instruction Name** | Renumber |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 2 |
| **Command Type** | Command |
| **Command Data** | New Number |
| **Reply Data** | Device Id |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Assigns new numbers to all the devices in the order in which they are connected. |

This command is usually sent to device number 0. When it is, the command data is ignored and all devices will renumber. The device closest to the computer becomes device number 1. The next device becomes number 2 and so on.

If sent to a device number other than 0, then that device will reassign itself the device number in the command data. Valid device numbers are 1-99 for version 6.05, and 1-254 otherwise.

**Note:** Renumbering takes about 1/2 a second during which time the computer must not send any further data. The device number is stored in non-volatile memory so you can renumber once and not worry about issuing the renumber instruction again after each power-up.

# Store Current Position - Cmd 16

| | |
|---|---|
| **Instruction Name** | Store Current Position |
| **Applies to** | All motorized devices |
| **Firmware Version** | 5.04 and up |
| **Command Number** | 16 |
| **Command Type** | Command |
| **Command Data** | Address |
| **Reply Data** | Address |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Saves the current absolute position of the device. |

Valid Address values are 0 through 15 specifying one of 16 possible registers in which to store the position. This command can only be executed when the device has been homed. This command is used in conjunction with the Return Stored Position (Command #17) and Move To Stored Position (Command #18) instructions. The positions stored in the position registers are non-volatile and will persist after power-down or reset. All position registers are cleared by the Restore Settings (Command #36) instruction.

# Return Stored Position - Cmd 17

| | |
|---|---|
| **Instruction Name** | Return Stored Position |
| **Applies to** | All motorized devices |
| **Firmware Version** | 5.04 and up |
| **Command Number** | 17 |
| **Command Type** | Command |
| **Command Data** | Address |
| **Reply Data** | Stored Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Returns the position stored in one of the 16 position registers for the device. |

Valid command data values are 0 through 15, specifying one of 16 possible registers from which to retrieve the position. This command is used in conjunction with the Store Current Position (#16) and Move To Stored Position (#18) commands. Positions stored in the position registers are non-volatile and will persist after power-down or reset. All position registers are cleared by the Restore Settings (#36) command.

# Move To Stored Position - Cmd 18

| | |
|---|---|
| **Instruction Name** | Move To Stored Position |
| **Applies to** | All motorized devices |

| Firmware Version | 5.04 and up |
|---|---|
| Command Number | 18 |
| Command Type | Command |
| Command Data | Address |
| Reply Data | Final Position |
| Safe to retry? | Yes |
| Returns Current Position? | Yes |
| Persistence | n/a |
| Summary | Moves the device to the stored position specified by the Command Data. |

Valid address values are 0 through 15, specifying one of 16 possible positions. This command is used in conjunction with the Store Current Position (#16) and Return Stored Position (#17) commands. This command does not send a response until the move has finished. All move commands are pre-emptive. If a new move command is issued before the previous move command is finished, the device will immediately move to the new position.

The target speed and acceleration during a move absolute instruction can be specified using Set Target Speed (Cmd 42) and Set Acceleration (Cmd 43) respectively.

This command may pre-empt, or be pre-empted by Move to Stored Position (Cmd 18), Move Absolute (Cmd 20), Move Relative (Cmd 21), Move at Constant Speed (Cmd 22) and Stop (Cmd 23).

# Move Absolute - Cmd 20

| Instruction Name | Move Absolute |
|---|---|
| Applies to | T-Series motorized devices |
| Firmware Version | 5.xx |
| Command Number | 20 |
| Command Type | Command |
| Command Data | Absolute Position |
| Reply Data | Final Position |
| Safe to retry? | Yes |
| Returns Current Position? | Yes |
| Persistence | n/a |
| Summary | Moves the device to the position specified in the Command Data in microsteps. |

The device begins to move immediately, and sends a response when the move has finished. The position must be between 0 and Maximum Position (specified by Set Maximum Position (cmd 44)), or an error code will be returned.

The target speed and acceleration during a move absolute instruction can be specified using Set Target Speed (Cmd 42) and Set Acceleration (Cmd 43) respectively.

All move commands are pre-emptive. If a new move command is issued before the previous move command is finished, the device will immediately move to the new position. This command may pre-empt, or be pre-empted by Move to Stored Position (Cmd 18), Move Absolute (Cmd 20), Move Relative (Cmd 21), Move at Constant Speed (Cmd 22) and Stop (Cmd 23).

# Move Relative - Cmd 21

| | |
|---|---|
| **Instruction Name** | Move Relative |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 21 |
| **Command Type** | Command |
| **Command Data** | Relative Position |
| **Reply Data** | Final Position |
| **Safe to retry?** | No |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Moves the device by the positive or negative number of microsteps specified in the Command Data. |

The device moves to a position given by its current position plus the value specified in the command data. The relative move command data in microsteps can be positive or negative. The final position must be between 0 and Maximum Position (specified by Set Maximum Position (cmd 44)), or an error code will be returned. The device begins to move immediately, and sends a response when the move has finished.

The target speed and acceleration during a move absolute instruction can be specified using Set Target Speed (Cmd 42) and Set Acceleration (Cmd 43) respectively.

All move commands are pre-emptive. If a new move command is issued before the previous move command is finished, the device will immediately move to the new position. If a Move Relative command is issued while the device is currently moving due to a previous command, the device will immediately set a new target position equal to the current position (at the instant the command was received) plus the specified relative position.

This command may pre-empt, or be pre-empted by Move to Stored Position (Cmd 18), Move Absolute (Cmd 20), Move Relative (Cmd 21), Move at Constant Speed (Cmd 22) and Stop (Cmd 23).

# Move At Constant Speed - Cmd 22

| | |
|---|---|
| **Instruction Name** | Move At Constant Speed |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 22 |
| **Command Type** | Command |
| **Command Data** | Speed |
| **Reply Data** | Speed |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |

| Persistence | n/a |
|---|---|
| Summary | Moves the device at a constant speed based on the value specified in the Command Data. |

This instruction specifies a direction and a speed to move, rather than a target position. When this instruction is issued the device will accelerate (at a rate determined by command #43 Set Acceleration) to the speed specified by the instruction data. The device will continue moving at this speed until a limit is reached or a pre-empting instruction is issued. Negative speeds cause retraction while positive speeds cause extension. Unlike the other movement commands, this command sends a response immediately without waiting for the move to finish.

The device may be set to return its position continuously during the move using the set mode command (#40)] bit 4. Position tracking is a reply-only command #8. If the device runs into zero position or maximum range, the device stops and the new position is returned via reply-only command #9.

This command may pre-empt, or be pre-empted by commands 18, 20, 21, 22 and 23.

For a spreadsheet that can be used to calculate speed setting values for any product see http://www.zaber.com/documents/ZaberSpeedSetting.xls. Alternatively you may use the formulas below.

Actual Speed

= Data * 9.375 * M mm/s or deg/s
= Data * 9.375 microsteps/s
= Data * 9.375 / R steps/s
= Data * 9.375 / (R x S) * 60 revolutions/min Motor rpm
= Data * 9.375 * L / (R x S) mm/s Linear devices only

where:

- Data is the value of the command data
- R (microsteps/step) is the microstep resolution (command 37)
- S (steps/revolution) is the number of steps per motor revolution
- M (mm or deg) is the microstep size
- L (mm or deg) is the distance of linear motion per motor revolution

Refer to product specifications for the distance corresponding to a single microstep or revolution.

For example, if a motor has 48 steps per revolution (S = 48), used with default resolution (R = 64), and Data is 2922, then the motor will move at a speed of approximately 535 revolutions per minute.

Valid data values are from (?512×R?1) to (512×R?1). Note that a value of zero will cause the device to decelerate to a stop and then send Limit Active (Cmd 9).

# Stop - Cmd 23

| Instruction Name | Stop |
|---|---|
| Applies to | T-Series motorized devices |

| | |
|---|---|
| **Firmware Version** | 5.xx |
| **Command Number** | 23 |
| **Command Type** | Command |
| **Command Data** | Ignored |
| **Reply Data** | Final Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Stops the device from moving by preempting any move instruction. |

This instruction can be used to pre-empt any move instruction. The device will decelerate to a stop. The reply data is the absolute position after stopping.

The device will decelerate at a rate specified by Set Acceleration (Cmd 43).

This command may pre-empt, or be pre-empted by Move to Stored Position (Cmd 18), Move Absolute (Cmd 20), Move Relative (Cmd 21), Move at Constant Speed (Cmd 22) and Stop (Cmd 23).

## Read Or Write Memory - Cmd 35

| | |
|---|---|
| **Instruction Name** | Read Or Write Memory |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 35 |
| **Command Type** | Command |
| **Command Data** | Data |
| **Reply Data** | Data |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Reads or writes a byte of non-volatile memory. |

128 bytes of memory are available for user data. For example, the user may want to save some custom data such as a serial number, a name string, or data that uniquely identifies a particular device. Data written is not cleared by power down or reset. The most significant bit of byte 3 specifies whether the instruction is a read (0) or a write (1). The least significant 7 bits of byte 3 specify the address to read/write (0 to 127). Byte 4 specifies the value to be written. Bytes 5 and 6 are ignored.

These settings are stored in non-volatile memory and will persist after power-down or reset.

## Restore Settings - Cmd 36

| | |
|---|---|
| **Instruction Name** | Restore Settings |
| **Applies to** | All Zaber devices except controllers |

| | |
|---|---|
| **Firmware Version** | 5.00 and up |
| **Command Number** | 36 |
| **Command Type** | Command |
| **Command Data** | Peripheral Id |
| **Reply Data** | Peripheral Id |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Restores the device settings to the factory defaults. |

This command should be issued with a Peripheral Id of 0 to return the device to factory default settings. This instruction is very useful for troubleshooting. If the device does not appear to function properly, it may be because some of the settings have been changed. This instruction will restore the settings to default values. For a table of default settings, see Appendix A. All settings affected by this instruction are stored in non-volatile memory and will persist after power-down or reset.

## Set Microstep Resolution - Cmd 37

| | |
|---|---|
| **Instruction Name** | Set Microstep Resolution |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 37 |
| **Command Type** | Setting |
| **Command Data** | Microsteps |
| **Reply Data** | Microsteps |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Changes the number of microsteps per step. |

This command sets the microstep resolution of a device.

This setting is stored in non-volatile memory and will persist after power-down or reset. Use Restore Settings (Cmd 36) to restore all non-volatile settings to factory default.

The default on most devices is 64. Available microstep resolutions are:

1, 2, 4, 8, 16, 32, 64, 128

All position data sent to or received from T-Series products is in units of microsteps. Note that when you change the microstep resolution, other position related settings are scaled automatically from current values to adjust for the new microstep size. The table below gives an example showing how other settings are affected when the microstep resolution is changed from 128 to 64:

| Setting | Before | After |
|---|---|---|

| | | |
|---|---|---|
| Target Speed * | 2922 | 1461 |
| Maximum Travel Range * | 280000 | 140000 |
| Current Position | 10501 ** | 5250 ** |
| Maximum Relative Move * | 20000 | 10000 |
| Home Offset * | 1000 | 500 |
| Acceleration * | 100 | 50 |

* The settings for these commands are saved in non-volatile memory.

** Note that if a number is divided by two, it is rounded down to the nearest whole number. The only exception to this is if acceleration would become 0 (because 0 for acceleration indicates infinite acceleration). If acceleration would become 0, it will instead be set to 1 which is the lowest acceleration possible.

# Set Running Current - Cmd 38

| | |
|---|---|
| **Instruction Name** | Set Running Current |
| **Applies to** | T-series motorized devices (excluding T-LSQ, T-LST, T-MCA) |
| **Firmware Version** | 5.xx |
| **Command Number** | 38 |
| **Command Type** | Setting |
| **Command Data** | Value |
| **Reply Data** | Value |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the desired current to be used when the device is moving. |

If your application does not require high torque, it is best to decrease the driving current to reduce power consumption, vibration, and motor heating. Trial and error should suggest an appropriate setting. If higher torque is required, it is generally safe to overdrive motors as long as they are not operated continuously. Motor temperature is typically the best indication of the degree to which overdriving can be employed. If the motor gets too hot to touch (>75°C), you should reduce the running current.

The current is related to the data by the formula:

Current = CurrentCapacity * 10 / CommandData

The range of accepted values is 0 (no current), 10 (max) - 127 (min). CurrentCapacity is the hardware's maximum capability of output current.

To prevent damage, some devices limit the maximum output current to a lower value. In that case the valid range is 0, Limit - 127. Current limits are listed under the device specifications.

Some devices limit the voltage rather than the current. In this case the same formula can be used by replacing Current and CurrentCapacity with Voltage and PowerSupplyVoltage.

For example, Suppose you connect a stepper motor rated for 420mA per phase to a T-CD2500. Reversing the equation above and using 420mA as Current gives:

CommandData

= 10 * CurrentCapacity / Current
= 10 * 2500mA / 420mA
= 59.5 (round to 60)

Therefore CommandData = 60.

# Set Hold Current - Cmd 39

| | |
|---|---|
| **Instruction Name** | Set Hold Current |
| **Applies to** | T-series motorized devices (excluding T-LSQ, T-LST, T-MCA) |
| **Firmware Version** | 5.xx |
| **Command Number** | 39 |
| **Command Type** | Setting |
| **Command Data** | Value |
| **Reply Data** | Value |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the desired current to be used when the device is holding its position. |

It is typical to run stepper motors at their rated current only during moves (for highest torque) and reduce the current when idle just to hold the position.

Typically the hold current can be set to around 25 - 50% of the running current. In some applications, the friction of the drive system alone is sufficient to hold the microstep position of the motor, and the hold current can be turned off completely. The hold current can be turned off by issuing the "Set Hold Current" instruction with data of 0.

When the device is moving, it applies running current to the motor. When the device stops moving, running current is applied for an additional 0.1 second before hold current is applied.

The current is related to the data by the formula:

Current = CurrentCapacity * 10 / CommandData

The range of accepted values is 0 (no current), 10 (max) - 127 (min). CurrentCapacity is the hardware's maximum capability of output current.

To prevent damage, some devices limit the maximum output current to a lower value. In that case the valid range is 0, Limit - 127. Current limits are listed under the device specifications.

Some devices limit the voltage rather than the current. In this case the same formula can be used by replacing Current and CurrentCapacity with Voltage and PowerSupplyVoltage.

For example, Suppose you connect a stepper motor rated for 420mA per phase to a T-CD2500. Reversing the equation above and using 420mA as Current gives:

CommandData

= 10 * CurrentCapacity / Current
= 10 * 2500mA / 420mA
= 59.5 (round to 60)

Therefore CommandData = 60.

# Set Device Mode - Cmd 40

| | |
|---|---|
| **Instruction Name** | Set Device Mode |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx since 5.04 |
| **Command Number** | 40 |
| **Command Type** | Setting |
| **Command Data** | Mode |
| **Reply Data** | Mode |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the Mode for the given device. |

This command allows setting several options. Each option is controlled by a single bit within the command data. Most software you will encounter, including most of our demo software, represents all 4 data bytes as a single decimal value rather than specifying each bit individually. To determine what decimal value to use requires a basic understanding of how the data is represented in binary. The command data may be considered as a single 32-bit binary value. The least significant bit is bit_0, the next is bit_1, the next is bit_2, and so on up to the most significant bit_31. Each bit may have a value of either 1 or 0.

The corresponding decimal representation of this 32-bit data is given by:

Decimal value = (bit_0 * 1) + (bit_1 * 2) + … + (bit_n * 2^n) + … + (bit_31 * 2^31)

Each bit controls a single mode option as described in the table below. To determine the data value to use with the Set Device Mode command, simply determine the desired value of each bit (1 or 0), and calculate the decimal value using the above formula. Note that not all 32 bits are currently used. Any unused or reserved bits should be left as 0.

For example, suppose you want all mode bits to be 0 except for bit_3 (disable potentiometer), bit_14 (disable power LED), and bit_15 (disable serial LED). The Set Device Mode instruction should be sent with data calculated as follows:

Command Data

= 2^3 + 2^14 + 2^15
= 8 + 16384 + 32768
= 49160

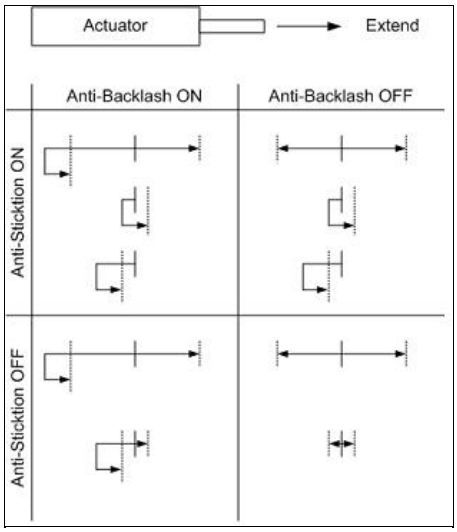
Note that each instance of the Set Device Mode command overwrites ALL previous mode bits. Repeated commands do not have a cumulative effect. For example, suppose you send a Set Device Mode command with data of 8 to disable the potentiometer. If you then send another Set Device Mode command with data of 16384 to disable the power LED, you will re-enable the potentiometer since bit_3 in the 2nd instruction is 0.

Most devices have a default mode setting of 0 (all bits are 0), however, there are some exceptions. See Appendix A of the user manual for a table of default settings.

| Bit_n | 2^n | Description |
|---|---|---|
| bit_0 | 1 | **Disable Auto-reply**<br>A value of 1 disables ALL replies except those to “echo”, “read”, “renumber”, and “return” commands. The default value is 0 on all devices. |
| bit_1 | 2 | **Enable Anti-backlash Routine**<br>A value of 1 enables anti-backlash. On negative moves (retracting), the device will overshoot the desired position by 640 microsteps (assuming 64 microsteps/step), reverse direction and approach the requested position from below. On positive moves (extending), the device behaves normally. Care must be taken not to crash the moving payload into a fixed object due to the 640 microsteps overshoot on negative moves. The default value is 0 on all devices. See note on anti-backlash and anti-sticktion below. * |
| bit_2 | 4 | **Enable Anti-sticktion Routine**<br>A value of 1 enables the anti-sticktion routine. On moves less than 640 microsteps (assuming 64 microsteps/step), the device will first retract to a position 640 microsteps less than the requested position and approach the requested position from below. Care must be taken not to crash the moving payload into a fixed object due to the 640 microsteps negative move. The default value is 0 on all devices. See section on anti-backlash and anti-sticktion below this table. * |
| bit_3 | 8 | **Disable Potentiometer**<br>A value of 1 disables the potentiometer preventing manual adjustment of the device. The default value is 0 on all devices. |
| bit_4 | 16 | **Enable Move Tracking**<br>A value of 1 enables the Move Tracking response during move commands. The device will return its position periodically (every 0.25 sec) when a move command is executed. The Disable Auto-Reply option above takes precedence over this option. The default value is 0 on all devices. Before firmware version 5.14, only Move at Constant Speed commands could generate tracking responses, now all move commands can. |
| bit_5 | 32 | **Disable Manual Move Tracking**<br>A value of 1 disables the Manual Move Tracking response during manual moves. The Disable Auto-Reply option above takes precedence over this option. The default value is 0 on all devices. |
| bit_6 | 64 | **Enable Message Ids**<br>A value of 1 enables Message Ids. In this mode of communication, only bytes 3 through 5 are used for data. Byte 6 is used as an Id byte that the user can set to any value they wish. It will |

| | | |
|---|---|---|
| | | be returned unchanged in the reply. Message Ids allow the users application to monitor communication packets individually to implement error detection and recovery. The default value is 0 on all devices. Prior to firmware version 5.06, this feature was called "Virtual Channels Mode" and did not behave reliably. We do not recommend enabling this mode of communications unless you have firmware version 5.06 or later. |
| bit_7 | 128 | **Home Status**<br>This bit is set to 0 automatically on power-up or reset. It is set automatically when the device is homed or when the position is set using command #45. It can be used to detect if a device has a valid position reference. It can also be set or cleared by the user. |
| bit_8 | 256 | **Disable Auto-Home**<br>A value of 1 disables auto-home checking. Checking for trigger of home sensor is only done when home command is issued. This allows rotational devices to move multiple revolutions without re-triggering the home sensor. |
| bit_9 | 512 | **Reverse Potentiometer**<br>A value of 1 reverses the direction of the travel when the potentiometer is used to control the device. This mode bit was introduced in firmware version 5.06. Prior to that it was not used. |
| bit_10 | 1,024 | **Reserved** |
| bit_11 | 2,048 | **Enable Circular Phase Microstepping**<br>Square phase microstepping is employed by default. A value of 1 enables circular phase microstepping mode. The differences are:<br><br>Circular Phase:<br>• constant torque<br>• smoothest operation<br>• better microstep accuracy<br>• only 70% torque (and lower power consumption)<br><br>Square Phase:<br>• non constant torque<br>• less smooth operation<br>• poorer microstep accuracy<br>• 100% torque achieved (and higher power consumption) |
| bit_12 | 4,096 | Reserved |
| bit_13 | 8,192 | Reserved |
| bit_14 | 16,384 | **Disable Power LED**<br>A value of 1 turns off the green power LED. It will still blink briefly, immediately after powerup. |
| bit_15 | 32,768 | **Disable Serial LED**<br>A value of 1 turns off the yellow serial LED. |

*Anti-backlash and Anti-sticktion routines are designed to compensate for backlash and sticktion. The solution to backlash is to always approach a position from the same direction. The solution to sticktion is to move the device far enough away from the final position to break free of sticktion before attempting the final move. The operation of the two features are dependent on each other, and the interaction of enabling one or both of the features is described in the diagram to the right.




For each setting scenario, the starting position is denoted by the solid vertical line and the final position is denoted by the dotted vertical line. There are four possible moves for each scenario: long move positive, long move negative, short move positive and short move negative. The arrows show the path that would be traversed for each scenario.

# Set Home Speed - Cmd 41

| | |
|---|---|
| **Instruction Name** | Set Home Speed |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx since 5.20 |
| **Command Number** | 41 |
| **Command Type** | Setting |
| **Command Data** | Speed |
| **Reply Data** | Speed |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the speed at which the device moves when using the "Home" command. |

When a home instruction is issued, the device will accelerate at a rate determined by the acceleration setting up to the speed determined by this command. The home speed can be set independently of target speed (command 42) so that for stages that move very fast, issuing the home command would use a slower home speed and prevent the stage from crashing into the home position.

For a spreadsheet that can be used to calculate speed setting values for any product see http://www.zaber.com/documents/ZaberSpeedSetting.xls. Alternatively you may use the formulas below.

Actual Speed

= Data * 9.375 * M mm/s or deg/s
= Data * 9.375 microsteps/s

= Data * 9.375 / R steps/s
= Data * 9.375 / (R x S) * 60 revolutions/min Motor rpm
= Data * 9.375 * L / (R x S) mm/s Linear devices only

where:

- Data is the value of the command data
- R (microsteps/step) is the microstep resolution (command 37)
- S (steps/revolution) is the number of steps per motor revolution
- M (mm or deg) is the microstep size
- L (mm or deg) is the distance of linear motion per motor revolution

Refer to product specifications for the distance corresponding to a single microstep or revolution.

For example, if a motor has 48 steps per revolution (S = 48), used with default resolution (R = 64), and Data is 2922, then the motor will move at a speed of approximately 535 revolutions per minute.

Valid data values are from 1 to (512×R?1).

# Set Target Speed - Cmd 42

| | |
|---|---|
| **Instruction Name** | Set Target Speed |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 42 |
| **Command Type** | Setting |
| **Command Data** | Speed |
| **Reply Data** | Speed |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the speed at which the device moves when using the "Move Absolute" or "Move Relative" commands. |

When a move absolute or move relative instruction is issued, the device will accelerate at a rate determined by the acceleration setting up to the speed determined by this command.

The target velocity may be changed on-the-fly even when the device is in the middle of a move. The device will automatically adjust the velocity, but still target the final position specified in the original move.

For a spreadsheet that can be used to calculate speed setting values for any product see http://www.zaber.com/documents/ZaberSpeedSetting.xls. Alternatively you may use the formulas below.

Actual Speed

= Data * 9.375 * M mm/s or deg/s

= Data * 9.375 microsteps/s
= Data * 9.375 / R steps/s
= Data * 9.375 / (R x S) * 60 revolutions/min Motor rpm
= Data * 9.375 * L / (R x S) mm/s Linear devices only

where:

- Data is the value of the command data
- R (microsteps/step) is the microstep resolution (command 37)
- S (steps/revolution) is the number of steps per motor revolution
- M (mm or deg) is the microstep size
- L (mm or deg) is the distance of linear motion per motor revolution

Refer to product specifications for the distance corresponding to a single microstep or revolution.

For example, if a motor has 48 steps per revolution (S = 48), used with default resolution (R = 64), and Data is 2922, then the motor will move at a speed of approximately 535 revolutions per minute.

Valid data values are from 0 to (512×R?1). In Firmware 5.21 and 5.22, a value of 0 is not allowed. In all other versions, target speed of 0 will cause Move Absolute/Relative and Move to Stored Position commands to return an error.

# Set Acceleration - Cmd 43

| | |
|---|---|
| **Instruction Name** | Set Acceleration |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 43 |
| **Command Type** | Setting |
| **Command Data** | Acceleration |
| **Reply Data** | Acceleration |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the acceleration used by the movement commands. |

When a movement command is issued, the device will accelerate at a rate determined by this command "Set Acceleration" up to a maximum speed determined by the target velocity. The acceleration may be changed on-the-fly even when the device is in the middle of a move. To determine the acceleration that will result from a given data value, the following formulas may be used:

Actual Acceleration

= 11250 * Data * M mm/s^2 or deg/s^2
= 11250 * Data microsteps/s^2
= 11250 * Data / R steps/s^2

Where:

- Data is the value specified in the Command Data
- M (mm or deg) is the microstep size
- R is the microstep resolution set in command #37 (microsteps/step)

The maximum value allowable is (512*R-1). This is the same as the maximum allowable data for velocity, which means that the device will reach maximum velocity immediately. If acceleration is set to 0, it is as if acceleration is set to (512*R-1). Effectively acceleration is turned off and the device will start moving at the target speed immediately.

# Set Maximum Position - Cmd 44

| | |
|---|---|
| **Instruction Name** | Set Maximum Position |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 44 |
| **Command Type** | Setting |
| **Command Data** | Range |
| **Reply Data** | Range |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the maximum position the device is allowed to travel to. |

Use this command to limit the range of travel to a value other than the default. Exercise caution when using this command, since it is possible to set the range to a value greater than the physical limits of the device.

A device within range of travel is not allowed to move above its Maximum Position. Valid values can be any number from 0 to 16777215.

The behaviour of this command depends on the firmware version:

5.01 - 5.20

Device movement behaviour when out of range is not well-defined.

5.21 - 5.22

The new Maximum Position cannot be less than the current position.

5.23 and up

If the device Current Position is out of range and above Maximum Position, the device is not allowed to move in the positive direction.

This setting is stored in non-volatile memory and will persist after power-down or reset.

NOTE: This command was previously named Set Maximum Range.

## Set Current Position - Cmd 45

| | |
|---|---|
| **Instruction Name** | Set Current Position |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 45 |
| **Command Type** | Setting |
| **Command Data** | New Position |
| **Reply Data** | New Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | Volatile |
| **Summary** | Sets the device internal position counter. |

This command override the internal position counter with a new position value specified by the user.

The position data is volatile and will not persist after power-down or reset.

The phase of the stepper motor is controlled by the least significant byte of the position, thus the device may move by +/- 2 full steps unless the new position corresponds to the true current position of the device. This command is useful if you want to turn off the system without losing position. Simply save the position in the controlling computer and turn off the hold current (Command 39) before powering down. After powering up, set the position back to the saved value and turn on the hold current. In this way you can continue without having to home the device. You have to turn off the hold current because when the power first comes on the position will default to the maximum range, and that may be out of phase with the motor's current position. If the hold current is on, it will force the motor into phase with the default position before you've had a chance to restore the current position.

In Firmware 5.21 and 5.22, the new Current Position must be equal or less than Maximum Position. See Set Maximum Position (Cmd 44) for more details on range settings and behaviour.

## Set Maximum Relative Move - Cmd 46

| | |
|---|---|
| **Instruction Name** | Set Maximum Relative Move |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 46 |
| **Command Type** | Setting |
| **Command Data** | Range |

| | |
|---|---|
| **Reply Data** | Range |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-volatile |
| **Summary** | Sets a limit on the number of microsteps the device can make for a Relative Move command. |

Use this command to limit the maximum range of travel for a relative move command. For example, if maximum relative move is set to 1000, and the user requests a relative move (#21) of 800, then the device will move 800 microsteps. However, if the user requests a relative move of 1200, then the device will reply with an error code. Most applications can leave this unchanged from the default.

This setting is stored in non-volatile memory and will persist after power-down or reset.

# Set Home Offset - Cmd 47

| | |
|---|---|
| **Instruction Name** | Set Home Offset |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 47 |
| **Command Type** | Setting |
| **Command Data** | Offset |
| **Reply Data** | Offset |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets the the new "Home" position which can then be used when the Home command is issued. |

When the home command is issued, the device will retract until the home sensor is triggered, then move forward until the home sensor is no longer triggered, then move forward by the Home Offset value (in microsteps) and resets the internal position (to 0 for most devices).

Note that the home offset command also changes the range settings of the device. For example, if the initial Home Offset is 0 and the Maximum Position is 500,000, and the user changes the Home Offset to 70,000, then the Maximum Position is automatically adjusted to be 430,000. However, changing the Maximum Position does not affect the home offset.

When a new Home Offset is specified, Maximum Position is adjusted to provide the same maximum location. However, the device will not be able to travel below its new home position unless it is homing.

This setting is stored in non-volatile memory and will persist after power-down or reset.

# Set Alias Number - Cmd 48

| | |
|---|---|
| **Instruction Name** | Set Alias Number |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 48 |
| **Command Type** | Setting |
| **Command Data** | Alias Number |
| **Reply Data** | Alias Number |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Sets an alternate device number for a device. |

This setting specifies an alternate device number for a device (in addition to its actual device number). By setting several devices to the same alias number, you can control a group of devices with a single instruction. When you send an instruction to an alias number, all devices with that alias number will execute the instruction and reply using their actual device numbers. To remove an alias, simply set the device's alias number to zero. Valid alias numbers are 0 to 99 for version 6.05, and 0 to 254 otherwise. To avoid confusion, it is best to choose an alias greater than the number of devices connected.

This setting is stored in non-volatile memory and will persist after power-down or reset.

# Set Lock State - Cmd 49

| | |
|---|---|
| **Instruction Name** | Set Lock State |
| **Applies to** | T-Series devices |
| **Firmware Version** | 5.xx since 5.07 |
| **Command Number** | 49 |
| **Command Type** | Setting |
| **Command Data** | Lock State |
| **Reply Data** | Lock State |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | Non-Volatile |
| **Summary** | Locks or unlocks all non-volatile settings. |

Sometimes it is desirable to lock all non-volatile settings to prevent them from being changed inadvertently. After changing all settings as desired, settings can be locked by setting the Lock State to 1. Subsequent attempts to change any non-volatile setting (e.g., Set Target Speed, command 42) will result in an error response with an error code of 3600 (settings locked). Note that the Set Lock State command does not apply to commands and settings that are specific to the joystick. Load Event Instruction and Set Axis Device Number for example, are unaffected by the Lock State.

How the Restore Settings instruction behaves when the settings are locked depends on the firmware version. In version 5.07 issuing a Restore Settings instruction while the settings are locked will result in an error

response with an error code of 3600 (settings locked). This behavior was found to confuse many customers so in version 5.08 and up, the behavior was changed such that regardless of the current lock state, issuing a Restore Settings instruction will always return setting values to factory default values and leave settings in an unlocked state.

Settings can also be unlocked by setting the Lock State to 0.

## Return Device Id - Cmd 50

| | |
|---|---|
| **Instruction Name** | Return Device Id |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 50 |
| **Command Type** | Read-Only Setting |
| **Command Data** | Ignored |
| **Reply Data** | Device Id |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Returns the id number for the type of device connected. |

See the Zaber support web site for a table of device ids for all Zaber products.

## Return Firmware Version - Cmd 51

| | |
|---|---|
| **Instruction Name** | Return Firmware Version |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 51 |
| **Command Type** | Read-Only Setting |
| **Command Data** | Ignored |
| **Reply Data** | Version |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Returns the firmware version installed on the device. |

A decimal is assumed before the last two digits. For example, 502 indicates firmware version 5.02.

# Return Power Supply Voltage - Cmd 52

| | |
|---|---|
| **Instruction Name** | Return Power Supply Voltage |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 52 |
| **Command Type** | Read-Only Setting |
| **Command Data** | Ignored |
| **Reply Data** | Voltage |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Returns the voltage level of the device's power supply. |

A decimal is assumed before the last digit. For example, a value of 127 indicates 12.7 V. Note that the internal voltage measurement is not very accurate. Don't be alarmed if the indicated voltage is slightly different from your measurements.

# Return Setting - Cmd 53

| | |
|---|---|
| **Instruction Name** | Return Setting |
| **Applies to** | All Zaber devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 53 |
| **Command Type** | Command |
| **Command Data** | Setting Number |
| **Reply Data** | Setting Value |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Returns the current value of the setting specified in the Command Data. |

Valid command data values are the command numbers of any "Set..." instruction. The device will reply using the command number of the specified setting (as if a command to change the setting had just been issued) but the setting will not be changed.

For example, command #48 is the "Set Alias" instruction. Therefore if you wish to return the current value of the alias number, simply send the Return Setting instruction with data of 48. The device will reply with command #48 and data equal to the setting value.

Since firmware version 5.21, this command also accepts the command numbers of any "Return..." instruction, such as command #50 "Return Device Id".

# Return Status - Cmd 54

| Instruction Name | Return Status |
|---|---|
| Applies to | T-Series motorized devices |
| Firmware Version | 5.xx |
| Command Number | 54 |
| Command Type | Read-Only Setting |
| Command Data | Ignored |
| Reply Data | Status |
| Safe to retry? | Yes |
| Returns Current Position? | No |
| Persistence | n/a |
| Summary | Returns the current status of the device. |

Possible status codes are as follows:

- 0 - idle, not currently executing any instructions
- 1 - executing a home instruction
- 10 - executing a manual move (i.e. the manual control knob is turned)
- 18 - executing a move to stored position instruction (FW 5.04 and up only)
- 20 - executing a move absolute instruction
- 21 - executing a move relative instruction
- 22 - executing a move at constant speed instruction
- 23 - executing a stop instruction (i.e. decelerating)

# Echo Data - Cmd 55

| Instruction Name | Echo Data |
|---|---|
| Applies to | All Zaber devices |
| Firmware Version | 5.04 and up |
| Command Number | 55 |
| Command Type | Command |
| Command Data | Data |
| Reply Data | Data |
| Safe to retry? | Yes |
| Returns Current Position? | No |
| Persistence | n/a |
| Summary | Echoes back the same Command Data that was sent. |

This command is useful for testing communication, similar to a network "ping".

# Return Current Position - Cmd 60

| | |
|---|---|
| **Instruction Name** | Return Current Position |
| **Applies to** | All motorized devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 60 |
| **Command Type** | Read-Only Setting |
| **Command Data** | Ignored |
| **Reply Data** | Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Returns the current absolute position of the device in microsteps. |

This is equivalent to issuing a Return Setting (#53) command with a command data value of 45 (Set Current Position).

# Reply-Only Reference

In general, a Zaber device will reply to an instruction using the same command number as the instruction itself. However, there are occasions (such as when the user turns a manual control knob) when the device may transmit data without first receiving a request from the controlling computer. This type of reply may be considered to be a triggered reply as opposed to a requested reply. In this case the device uses a “reply-only” command number to distinguish the reply from those requested by the controlling computer. The meanings of these replies and their corresponding data are given below.

## Move Tracking - Cmd 8

| | |
|---|---|
| **Instruction Name** | Move Tracking |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 8 |
| **Command Type** | Reply |
| **Command Data** | n/a |
| **Reply Data** | Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Indicates to the user that the device has been set to a position tracking mode and given a move instruction. |

Move Tracking has been enabled (see Set Device Mode (Cmd 40)) and device has been given a move instruction. In this mode, the device sends this reply every 0.25 seconds updating the current absolute position (in microsteps) during any move.

## Limit Active - Cmd 9

| | |
|---|---|
| **Instruction Name** | Limit Active |
| **Applies to** | All motorized devices |
| **Firmware Version** | 5.00 and up |
| **Command Number** | 9 |
| **Command Type** | Reply |
| **Command Data** | n/a |
| **Reply Data** | Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | Indicates to the user that the device has reached one of the limits of travel. |

This response from a device indicates that a “move at constant speed” command has finished. Generally this is because the device reached one of the limits of travel (either the minimum position or maximum position), but it also occurs if the device is ordered to move at constant speed zero.

## Manual Move Tracking - Cmd 10

| | |
|---|---|
| **Instruction Name** | Manual Move Tracking |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 10 |
| **Command Type** | Reply |
| **Command Data** | n/a |
| **Reply Data** | Position |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | Yes |
| **Persistence** | n/a |
| **Summary** | A reply that is sent when the manual control knob is turned. |

Manual Move Tracking has been enabled (see Set Device Mode (Cmd 40)) and the device has been moved manually (the knob is turned). In this mode, the device sends this reply every 0.25 seconds updating the current absolute position (in microsteps) during any move.

## Error - Cmd 255

| | |
|---|---|
| **Instruction Name** | Error |
| **Applies to** | T-Series motorized devices |
| **Firmware Version** | 5.xx |
| **Command Number** | 255 |
| **Command Type** | Reply |
| **Command Data** | n/a |
| **Reply Data** | Error Code |
| **Safe to retry?** | Yes |
| **Returns Current Position?** | No |
| **Persistence** | n/a |
| **Summary** | Indicates to the user that an error has occurred. |

This reply indicates that an error has occurred. The error code returned in the data indicates the type of error. The device may send an error code as a reply to an invalid instruction, or it may autonomously send an error code as a triggered reply (i.e. not in response to an instruction). The error code is typically the command number of the instruction that caused the error, but not always.

### Error Codes

| Code | Name | Description |
|---|---|---|
| 1 | Cannot Home | Home - Device has traveled a long distance without triggering the home sensor. Device may be stalling or slipping. |
| 2 | Device Number Invalid | Renumbering data out of range. |
| 14 | Voltage Low | Power supply voltage too low. |
| 15 | Voltage High | Power supply voltage too high. |
| 18 | Stored Position Invalid | The position stored in the requested register is no longer valid. This is probably because the maximum range was reduced. |
| 20 | Absolute Position Invalid | Move Absolute - Target position out of range. |
| 21 | Relative Position Invalid | Move Relative - Target position out of range. |
| 22 | Velocity Invalid | Constant velocity move. Velocity out of range. |
| 36 | Peripheral Id Invalid | Restore Settings - peripheral id is invalid. Please use one of the peripheral ids listed in the user manual, or 0 for default. |
| 37 | Resolution Invalid | Invalid microstep resolution. Resolution may only be 1, 2, 4, 8, 16, 32, 64, 128. |
| 38 | Run Current Invalid | Run current out of range. See command 38 for allowable values. |
| 39 | Hold Current Invalid | Hold current out of range. See command 39 for allowable values. |
| 40 | Mode Invalid | Set Device Mode - one or more of the mode bits is invalid. |
| 41 | Home Speed Invalid | Home speed out of range. The range of home speed is determined by the resolution. |
| 42 | Speed Invalid | Target speed out of range. The range of target speed is determined by the resolution. |
| 43 | Acceleration Invalid | Target acceleration out of range. The range of target acceleration is determined by the resolution. |
| 44 | Maximum Range Invalid | The maximum range may only be set between 1 and the resolution limit of the stepper controller, which is 16,777,215. |
| 45 | Current Position Invalid | Current position out of range. Current position must be between 0 and the maximum range. |
| 46 | Maximum Relative Move Invalid | Max relative move out of range. Must be between 0 and 16,777,215. |
| 47 | Offset Invalid | Home offset out of range. Home offset must be between 0 and maximum range. |
| 48 | Alias Invalid | Alias out of range. |
| 49 | Lock State Invalid | Lock state must be 1 (locked) or 0 (unlocked). |
| 53 | Setting Invalid | Return Setting - data entered is not a valid setting command number. Valid setting command numbers are the command numbers of any "Set ..." instructions. |
| 64 | Command Invalid | Command number not valid in this firmware version. |
| 255 | Busy | Another command is executing and cannot be pre-empted. Either stop the previous command or wait until it finishes before trying again. |
| 1600 | Save Position Invalid | Save Current Position register out of range (must be 0-15). |
| 1601 | Save Position Not Homed | Save Current Position is not allowed unless the device has been homed. |

| | | |
|---|---|---|
| 1700 | Return Position Invalid | Return Stored Position register out of range (must be 0-15). |
| 1800 | Move Position Invalid | Move to Stored Position register out of range (must be 0-15). |
| 1801 | Move Position Not Homed | Move to Stored Position is not allowed unless the device has been homed. |
| 2146 | Relative Position Limited | Move Relative (command 20) exceeded maximum relative move range. Either move a shorter distance, or change the maximum relative move (command 46). |
| 3600 | Settings Locked | Must clear Lock State (command 49) first. See the Set Lock State command for details. |
| 4008 | Disable Auto Home Invalid | Set Device Mode - this is a linear actuator; Disable Auto Home is used for rotary actuators only. |
| 4010 | Bit 10 Invalid | Set Device Mode - bit 10 is reserved and must be 0. |
| 4012 | Home Switch Invalid | Set Device Mode - this device has integrated home sensor with preset polarity; mode bit 12 cannot be changed by the user. |
| 4013 | Bit 13 Invalid | Set Device Mode - bit 13 is reserved and must be 0. |

# Manual Control

Turning the potentiometer knob will make the device move. It is not necessary to have a computer connected to use the device in manual mode. However, without a computer connected you have no means to initialize the device with a starting position. Therefore you must retract it completely after each power up in order to home the device. You will not be able to extend it fully until you have first retracted it completely to set the home position. Clockwise rotation of the knob produces positive motion (extension) and counter clockwise rotation produces negative motion (retraction).

The speed of retraction or extension will be directly related to the amount to which the knob is turned from its center detent position (turning the knob to its center position will stop the device from moving). During manual moves, the device constantly returns its position so a computer can track the position even when you are controlling the device manually. Manual control can be disabled with a mode setting.

The speed of the potentiometer-controlled manual moves is also determined by the Set Target Velocity command (#42). When the knob is fully turned to either limit, the device will move at the maximum velocity set by command #42. Any intermediate position of the knob will move the device at a velocity proportional to how far the knob has been turned. If the velocity is set to be very small, it may appear that turning the knob produces no movement. The way to verify that the actuator is moving is to connect it to a computer and monitor the replies back to the computer.

During operation if the actuator is extended or retracted against a force greater than its thrust capability the device will stall and there will be “missed steps”. This can result in an apparent malfunction in that the device believes its position to be other than it actually is and will not extend or retract the actuator past a given position. Without connecting a computer to home the device or set its position, the only solution is to retract the actuator until it activates the internal home switch, which will automatically zero the device at the home position. A problem arises if the device incorrectly believes its position to be zero since it will not retract to the home position. In this case you must disconnect and reconnect the power before manually homing the device.

# LED functions

The green LED is on whenever there is power to the device. A constant blinking (2Hz) of the green LED indicates that the power supply is out of range. The green LED may be disabled by a mode bit (see command #40)

The amber or red LED flashes when there is traffic packet on the RS232 line. It also stays on while the device is moving. When the potentiometer is turned, the amber or red LED will blink at a rate that is proportional to how far the potentiometer is turned. The farther the knob is turned away from center detent, the faster the LED will blink. If the device reaches either the home or the away limit while under manual control, the LED will blink slowly to indicate that the device cannot move any farther.

# Troubleshooting T-Series Motion Devices

| Symptom | Check |
|---|---|
| After the device finishes a move command, it starts moving again on its own. | The potentiometer knob is probably not centered. Make sure the knob is set to the middle position. You can feel a slight detent in the middle of its rotation. |
| Device starts turning as soon as the power is on or amber LED blinks all the time | The potentiometer is probably not centered. Turn the knob slowly until you feel the center detent. |
| Nothing happens when I turn the potentiometer knob either way. | Manual control may have been disabled. Issue the Restore Settings Instruction (command 36) or enable the potentiometer using the Set Device Mode Instruction (command 40). |
| Nothing happens when I send a move instruction. | The device needs to be "homed" first. You must issue the Home Instruction (command 1) after power up to home the device. |
| When I try to move the device it just makes a noise and vibrates. | There may be too much load that the device is trying to push. The actuator stalls in this situation. Try removing the load and see if the problem goes away. You can achieve higher thrust or torque by lowering the speed of the device using the Set Target Speed Instruction (command 42). |
| Stall condition: The device makes noise but does not move. | Try removing all external loads. If the device now extends and retracts normally, the problem is excessive load. Try to reduce the load or change step time and acceleration parameters to ensure the load is less than the maximum thrust. If a linear motion device is stalled in its fully extended position and remains stalled without any external load applied it means the lead screw has been over extended and is stuck. You can usually get the lead screw unstuck by pushing on it after issuing the home command. If a device is stalled (with no external load) in a position that is not fully extended then it requires servicing. |
| There is no communication between the computer and my Zaber device, the amber light does not come on or flash. | There are several things you should check:<br><br>1. Make sure you have the correct serial port selected. Try selecting other serial ports in the software. To ensure that your serial port is working properly, you can connect the send and receive pins of the serial adaptor without connecting any devices. Use a wire to connect pins 2 and 6. Your software should echo a reply every time you send an instruction. You can view the pinout diagram here.<br>2. Check the baud rate, hand shaking, parity, stop bit, etc. when configuring your serial communications software. The required settings are listed in the Control Through The RS-232 Serial Port section above. Also make sure that your serial port is not configured with a termination character (it often defaults to linefeed).<br>3. Make sure there are no bent pins in the ends of all the data cables<br>4. Make sure the power supply is working. Check the voltage over the DC plug to see if it measures close to the rated voltage for the supply.<br>5. Make sure the power plug is plugged in all the way. If your device has LEDs, the green LED should light. |

| | |
|---|---|
| | 6. If your device has a manual control knob, make sure it is centered (you should feel a detent in the center position).<br>7. If your computer is a laptop running on batteries, try plugging in the power. Some laptops disable the serial ports when running on batteries.<br>8. Make sure you do not have a null modem adaptor or cable in the line.<br>9. The serial to mini-din adaptor comes in many varieties and many have different pin connections. Check the adaptor for continuity on the proper pins by consulting the adaptor pin-out diagram above.<br>10. If you encounter the problem when trying to control the device with your own software, try using one of the demo programs from our website to verify that the hardware is functioning properly. |
| My device is behaving strangely. It responds to some commands as expected but not to others. | 1. Send a Restore Settings Instruction (command 36). A setting might have been inadvertently changed. If you have a T-MCA or T-CD series stepper motor controller, note that a data value should be entered with the Restore Default Settings Instruction corresponding to the Peripheral Id of the device you are controlling.<br>2. Your computer might be set to Unicode. This is common for languages that use non-Latin based characters. Go to Control Panel/Regional and Language Options/Advanced. Select a language for non-unicode programs. This should be English or another Latin based character language. |
| The device is moving very slowly. It used to behave differently. | The speed settings may have been changed inadvertently. Send a Restore Settings Instruction (command 36). |
| The device is not communicating or responding to computer control. The yellow LED may be blinking. | If the actuator has a manual control knob, make sure the knob is centered. Turn it back and forth until you feel a click or detent. Leave the knob at the center detent position. Then turn device off and on, and try again. |
| Green LED does not come on | Check the A/C wall plug, the voltage adaptor and its connection to the device. If the power is coming over the data cable, check the mini din connector for bent or broken pins.<br><br>The amber light should turn off. |
| Green LED flashes | The power supply voltage is outside the specified range for your device. It may either be too low or too high. Some unregulated adaptors may produce voltages significantly in excess of their rated values. If the number of devices connected on a single power supply exceeds its current capability, the voltage may drop below the required minimum voltage. You may experience this problem when many motors on a single supply move together. The load may exceed the maximum current available, causing the voltage to drop too low. If you experience this problem with a single device on a single unregulated supply rated at over 300 mA, then the problem is probably that the supply voltage is too high. |
| Turning the potentiometer causes no motion. | You may be at the end of travel. This can happen due to missed steps even if the device does not appear to be fully extended. Turn the knob the other way. If the device makes noise but does not move you may be in a stall condition (especially if the device |

| | |
|---|---|
| | appears to be fully extended). See “Stall Condition” below. The amber light should blink when turning the knob, if not, try turning the power off and then on again. You may also have set the Target Velocity (command #42) so low that it doesn’t produce any visible motion. Try using command #36 to reset the device to default settings and try again. |
| The amber light comes on briefly when sending a command, but the device does not move and does not return codes. | Check baud rate, hand shaking, parity, stop bit, etc. Make sure that your software does not transmit any control characters like line feed, spaces or something else. The device numbers may not be what you think they are. Issue a renumber command, make sure that the computer does not transmit anything else while the devices renumber. Check that you transmit 6 bytes and that the device number and command are valid. If you encounter the problem when trying to control the device with your own software, try using one of the demo programs from our website to verify that the hardware |
| The device does not send replies but otherwise works. | If you encounter the problem when trying to control the device with your own software, try using a demo program from our website to verify that the hardware is functioning properly. Make sure that the receiving part of your code or commercial package is correct. Check baud rate, etc. Check connectors for bent or broken pins. |
| The device sometimes returns fewer than 6 bytes. | This problem usually indicates a problem with the settings for your serial port. Some serial ports are set to automatically recognize and remove specific control characters such as carriage returns when they appear in the RS232 receive buffer. When this happens, it appears as though the device has not sent enough bytes, but really the controlling computer has just removed some before you could read them. You will need to change the serial port settings to fix the problem. |
| Poor repeatability or the device does not extend or retract smoothly or makes louder than normal noise during travel. | You may be skipping steps. When skipping, the device will lose position in increments of 4 full steps (not microsteps). This condition happens if the thrust needed is more than the thrust available from the device. Check that the force on the device is less than the maximum thrust. Check the voltage using the voltage command. Voltage less than the rated or recommended voltage will reduce the device’s maximum thrust. Try a slower target velocity (command #42) as stepper motors produce more thrust when moving slowly. Lead screw conditions greatly affect the performance of linear motion devices. Dirt, damaged threads, no grease or too heavy grease will degrade performance and may contribute to a stall. A black residue appears on the lead screw after extended use. This can increase friction and reduce thrust. Clean the screw and re-grease it. In general if you try to move a large payload or have a large static axial load (like lifting something vertically) you will have more problems. For vertical motion the use of a counterweight, spring or rubber band can help reduce the static load and improve the performance of the device. The default value of the acceleration and target velocity are good for small to medium loads and medium speeds. For very light loads and higher speeds, or heavy loads at lower speeds, these parameters can be tuned. Trial and error is the best tuning technique. |
| The device has repeatability errors smaller than 4 full steps. | If you're not skipping steps, friction may still cause some variation when returning to a position. Depending on the exact cause, there are a couple of device modes that can reduce these errors. See the Set Device Mode command's sections on the anti-backlash and anti-sticktion routines. |
| A linear motion device extends and retracts smoothly but will not retract to the home (zero) position. | The device will not retract below what it believes to be the zero position. If the device has missed steps due to a previous stall condition or if the device has been set to an incorrect position, the device may incorrectly believe it is at the zero position. You can solve the problem by issuing the home command, or by turning the device on and off and manually homing it. |
| | |

| | |
|---|---|
| The device does not behave as expected when software sends it a series of commands. | It can be challenging to track down problems in a complicated script or other software. In addition to the standard techniques of debugging any software, it's also helpful to see exactly what bytes are being sent and received on the serial port. There are several tools available to display the raw data from a serial port, and stackoverflow.com has a list of some. |

# Warranty and Repair

For Zaber's policies on warranty and repair, please refer to the Ordering Policies

## Standard products

Standard products are any part numbers that do not contain the suffix ENG followed by a 4 digit number. Most, but not all, standard products are listed for sale on our website. All standard Zaber products are backed by a one-month satisfaction guarantee. If you are not satisfied with your purchase, we will refund your payment minus any shipping charges. Goods must be in brand new saleable condition with no marks. Zaber products are guaranteed for one year. During this period Zaber will repair any products with faults due to manufacturing defects, free of charge.

## Custom products

Custom products are any part numbers containing the suffix ENG followed by a 4 digit number. Each of these products has been designed for a custom application for a particular customer. Custom products are guaranteed for one year, unless explicitly stated otherwise. During this period Zaber will repair any products with faults due to manufacturing defects, free of charge.

## How to return products

Customers with devices in need of return or repair should contact Zaber to obtain an RMA form which must be filled out and sent back to us to receive an RMA number. The RMA form contains instructions for packing and returning the device. The specified RMA number must be included on the shipment to ensure timely processing.

# Email Updates

If you would like to receive our periodic email newsletter including product updates and promotions, please sign up online at www.zaber.com (news section). Newsletters typically include a promotional offer worth at least $100.

# Contact Information

Contact Zaber Technologies Inc by any of the following methods:

| | |
|---|---|
| **Phone** | 1-604-569-3780 (direct)<br>1-888-276-8033 (toll free in North America) |
| **Fax** | 1-604-648-8033 |
| **Mail** | #2 - 605 West Kent Ave. N., Vancouver, British Columbia, Canada, V6P 6T7 |
| **Web** | www.zaber.com |
| **Email** | Please visit our website for up to date email contact information. |

The original instructions for this product are available at http://www.zaber.com/wiki/Manuals/T-LSM-SV.

# Appendix A: Default Settings

see website

# Appendix B: Device Specifications

For complete device specifications for T-LSM-SV stages please see our website.